International Journal of Informatics and Communication Technology (IJ-ICT)
Vol. 14, No. 1, April 2025, pp. 141~152
ISSN: 2252-8776, DOI: 10.11591/ijict.v14i1.pp141-152

Memory management of firewall filtering rules using modified tree rule approach

Dhwani Hakani, Palvinder Singh Mann
School of Engineering and Technology, Gujarat Technological University, Ahmedabad, India

Article Info

Article history:

Received Apr 1, 2024
Revised Oct 24, 2024
Accepted Nov 19, 2024

Keywords:

Cloud security
Conflicts resolution
Correlation
Firewall rules
Redundancy
Rule reordering
Shadowing

ABSTRACT

Firewalls are essential for safety and are used for protecting a great deal of private networks. A firewall's goal is to examine every incoming and outgoing data before granting access. A notable kind of conventional firewall is the rule-based firewall. However, when it comes to job performance, traditional listed-rule firewalls are limited, and they become useless when utilized with some networks that have extremely big firewall rule sets. This study proposes a model firewall architecture called "Tree-Rule Firewall," which has benefits and functions effectively in large-scale networks like "cloud." In order to improve cloud network security, this study suggests a modified tree rule firewall (MTRF cloud) that eliminates rule discrepancies. For the matching firewall policy, this work creates a tree rule firewall. There are no duplicate rules created by the proposed improved tree rule firewall. Also, memory utilization of different size rules is compared

This is an open access article under the CC BY-SA license.

Corresponding Author:

Palvinder Signh Mann
School of Engineering and Technology, Gujarat Technological University
Ahmedabad, Gujarat, India
Email: ojugo.arnold@fupre.edu.ng

1. INTRODUCTION

Since firewalls are intended to avoid hostile attacks from infiltrating computer networks and shield websites from intrusion, they are a necessary security mechanism for connecting to current networks. However, if the defined rules do not cover all activity occurring on the networks, the level of security provided by firewalls is gradually decreased. For this reason, creating thorough firewall rules is crucial to network security. A firewall rule is essentially made up of six conditional statements [1] that are used to determine whether data, such as source and destination IP addresses, source and destination ports, protocols, and actions, can pass through or not. The intricacy of an organization's policy determines how many firewall rules are needed. Rule anomalies grow in number in tandem with the number of regulations. Theoretically, any two rules that overlap yet have distinct actions might result in a firewall rule anomaly. For example, the first control may provide Internet access for everyone in the firm, while the second rule prohibits Internet surfing by anybody working for the company. The hardware and software components used in network transactions are known as network systems, and they include making appointments, sending and receiving emails, reading news, keeping and distributing private papers, purchasing as well as learning [2]. In terms of computers, networks. Security is essential to safeguarding sensitive data of users and the network as a whole [3]. An electronic firewall is network system that manages and keeps an eye on the network traffic, including inbound and outbound, in compliance with includes guidelines and protocols for security [4], [5]. A barrier serves as the gate intended to identify unapproved access over a network that is not reliable [6]. Identifying anomalies in cloud firewalls remains a challenging area with significant research gaps. Addressing these gaps requires developing scalable, context-aware, real-time, and automated anomaly

detection solutions that are tailored to the unique characteristics of cloud environments. Additionally, solutions must handle encrypted traffic, reduce false positives and negatives, and integrate across multi-cloud and multi-tiered architectures. These gaps represent opportunities for both academic research and practical innovation in the field of cloud security.

The security of information technology is a modern, essential idea that is relevant in every way, especially in light of the Internet's information flow. The investigation of vulnerabilities in systems that are vulnerable to compromise is a critical problem for any organisation, whether public or private, given the increasing interest in cybersecurity. With nearly 100% success rates, it is now feasible to mitigate the effects of attacks on existing systems by utilising AI algorithms for the purpose of screening requests and determining their maliciousness. This entails maximising security even for altered data, which would be far more challenging for a human expert to find. Since firewalls are designed to keep hostile attacks out of computer networks and to protect websites from intrusion, they are a necessary security measure for connecting to contemporary networks. However, in the event that the rules provided do not cover every network activity, the firewall's protective measures are gradually reduced. Thus, putting strong firewall rules into place is essential for network security. Six conditional statements make up a firewall rule. These statements specify which data, such as the IP addresses of the source and destination, ports, protocols, and actions, may and cannot move (Blocked).

2. LITERATURE REVIEW

We go over rule anomaly checking and resolution in this section. The first study to propose the rule anomaly model for firewall rules with a corresponding verification mechanism was in [7]. In this, it is developed the rule anomaly within a single firewall as well as between firewalls in 2003 after studying the interactions between firewall rules [8]. These researchers suggested a verification technique based on the state transit diagram and illustrated a policy tree for each firewall rule. In [7], it is developed an anomaly checking technique and modelled firewall rules as binary decision diagrams (BDDs) based on their research. A conventional anomaly resolution system was presented by authors in [9], whose key concept is concurrently maintaining input and output rule set. The presence and confluence of the answer were also examined by these scholars. Firewall rules were suggested to be divided into a collection of meta-rules by author in [10], who then composed these meta-rules to accomplish fine-grain rule administration.

Furthermore, comparable research on additional safety functions has also been noted. In [11], for example, noted that an IPsec-based VPN had the same issues. Furthermore, in [12] examined the conflict between firewalls also VPNs (also known as network address translators) as well as suggested a matching checking technique. According to their earlier research, authors introduced a BDD-based the OpenFlow protocol flow-table checking approach [13] with the advent of SDN. This method is flexible enough to be used with a single the OpenFlow protocol switch. Subsequently, in [14], it concentrated on the OpenFlow switch's anomaly issues and suggested employing a hierarchy-based flow-table (HFT) and employing a tree structure to make judgements for every individual packet. After a year, using their previous work, SDN controller API was developed by these researchers and approved for usage by network administrators in a single network domain [15]. In order to address two primary issues, these researchers proposed an organisational share tree to categorise the management level: i) removing the control planes from the overall view of the network, and ii) addressing conflicts between various users. In response to their work, auhtor in [16] introduced an interval forests model for expeditious anomaly testing and privilege management utilising the prior sharing tree. In a security functional chain, there isn't currently a method in place to resolve anomalies pertaining to various security rules.

Therefore, we suggest a novel technique to address the issue in light of the research done by author. To prevent time-wasting processes, we create a priority-based approach that draws inspiration from the designs of author in [17] and in [18]. However, instead of being employed for SDN flow table construction, our suggested approach is used for addressing anomalies pertaining to generic security rules. Table 1 explains memory utilization of firewall rules.

Table 1. Memory utilization by different number of firewall rules

Rules #	Memory required (MB)	Protocol
100	518	TCP
200	956	TCP
400	1067	TCP
600	2015	TCP
800	3096	TCP
1000	6062	TCP
1200	11050	TCP

A firewall policy's standard format for packet filtering rules is as follows:
<order><protocol><src_ip><src_port><dst_ip><dst_ port><action>

3. IDENTIFICATION OF ANOMALIES

3.1. Shadowing anomaly

The shadowing phenomenon, as its name implies, occurs when a particular rule takes on the A rule is referred to be a shadow rule of a preceding rule when its predicate component matches the premise element of that previous rule but its actions are different. As in Figure 1, it shows Space for resolution of anomalies.

$$\left(\forall x, (R_a(x) \supseteq R_b(x))\right) \wedge (R_a(action) \neq R_b(action))$$

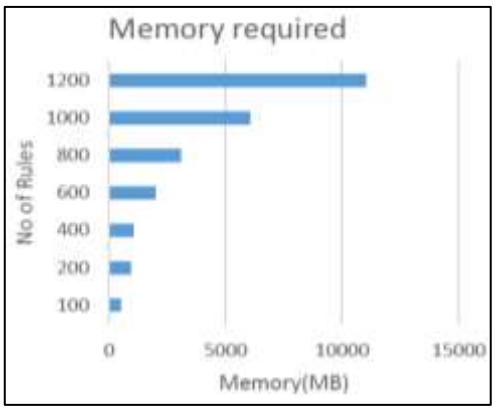

Figure 1. Space for resolution of anomalies

3.2. Correlation anomaly

Two rules are said to be closely correlated when they have different action fields, some of the initial rule's predication fields are the same as or smaller compared to the relate prediction fields in the rule that follows, and the remaining predicate fields in the subsequent rule are a superset of the predicate fields in the first rule.

3.3. Generalization anomaly

The other rule is called a generalisation of the first rule when both have different action values but every one of the subsequent rule's predicated fields match all the original rule's predicate fields.

$$\left(\forall x, (R_a(x) \subseteq R_b(x))\right) \wedge (R_a(action) \neq R_b(action))$$

It is not the same as observing, despite the similarities. Unlike generalisation, which occurs when another policy absorbs the superordinate policy, shadowing occurs when the previous policy matches or includes a subordinate policy. The generalisation anomaly arises when two rules have different actions and every packet discovered by one of them is a subset of every packet match by the other.

3.4. Redundancy anomaly

Eliminating a redundant rule has no impact on a security policy since it executes the same action on a single packet as another rule. Regulate Ry is unnecessary in identifying Rx if Rx happens before Ry within the sequence, Ry is a part of or exactly the same as Rx, and the activities are equivalent.

$\forall x: R_a(x) \cap R_b(x) \neq \emptyset$

$Where\ x \in \{protocol, SrcIP, DestIP, SrcPort, DestIP, action\}$

$If\ R_a(proto) \cap R_b(proto) \neq \emptyset\ and\ R_a(IPaddr) \cap R_b(IPaddr)$

$\neq \emptyset\ and\ R_a(port) \cap R_b(port) \neq \emptyset\ and\ R_a(action) \cap R_b(action)$

$\neq \emptyset\ then\ R_a\ is\ redundent\ to\ R_b$

4. ANALYSIS AND DETECTION OF INTRAFIREWALL ANOMALIES

The process of classifying packets involves matching each one in turn against firewall rules till a match is obtained. The hierarchy between the rules is not important if they are self-contained. Firewall rules frequently have one firewall rule connected to another even if they make distinct choices. In this instance, conflicts and the ensuing order sensitivity have logically intertwined the rules within a firewall policy.

Thus, the existence if two or more filters that may match the same packet or the existence of an exception that could never match a packet that crossed the firewall is classified as an intra-firewall policy anomaly. The rule configuration anomaly falls into one of three categories, as defined by the concept of policy anomaly: conflict, incompleteness, and redundancy.

r1:F1∈[0,7] ∧ F2∈[4,6] →accept,
r2:F1∈[0,8] ∧ F2∈[3,8] →discard

The reason why the two rules, r1 and r2, are at odds is that certain packets have fields that fulfil both of their predicates (for instance, a packet with F1∈[0,7] ∧ F2∈[3,8] can satisfy both predicates), and these two rules make different conclusions. Consequently, it becomes crucial to consider how these two rules relate to each other in the order of rules. It's probably a consistency mistake that rules both r1 and r2 are in the wrong sequence.

5. ANALYSIS AND DETECTION OF INTERFIREWALL ANOMALIES

Generally speaking, if any two firewalls on a connection filter the same data differently, there may be an inter-firewall anomaly. We assume that traffic is moving from domains D1 to domain D2, as shown in Figure 2. The most upwards firewall (FW1) is the one that is closest to the stream source domain (D1), and the most down firewall (FWn) is the one that is closest to the flow target domain (D2).

There may be anomalies between the rules of several firewalls even if none of the network's firewall policies include the rule abnormalities. For instance, a downstream firewall may allow traffic that an upstream firewall is blocking, or vice versa. According to [19], an anomaly occurs for any traffic going from domain D1 to domain D2 if any of the following circumstances is true, utilizing the network model depicted in Figure.

- The most down stream firewall accepts traffic that was previously refused through any of the subsequent upland firewalls;
- The most upstream firewall allows traffic that has been banned by all of the following firewalls; and
- The downstream firewall rejects traffic that was previously denied by the most powerful upstream firewall.

Any traffic that is permitted by the downward firewall must be simultaneously admitted by all upstream firewalls for the traffic to reach its destination.

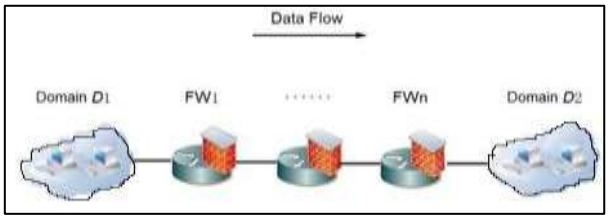

Figure 2. Inter firewall anomalies example with domains

6. LIMITATIONS

Intra-firewall anomaly detection refers to identifying unusual or suspicious activities within a firewall, such as deviations from expected traffic patterns or violations of security policies. While it is a critical component of network security, this approach has several limitations that can reduce its effectiveness. Below are some key limitations of intra-firewall anomaly detection:

1. High rate of false positives
   Explanation: Anomaly detection systems can often flag legitimate traffic as suspicious because they focus on deviations from normal patterns, which might include non-malicious changes (e.g., a new service deployment or temporary traffic spikes). This can overwhelm security teams with alerts, leading to alert fatigue and making it difficult to focus on actual threats.

Impact: Security teams may spend time chasing down benign events, slowing down response to real security incidents and lowering overall operational efficiency.

2. False negatives and missed threats
   Explanation: Anomaly detection systems are typically designed to detect unknown or novel threats. However, sophisticated attackers can disguise their activities to appear as normal traffic (e.g., by mimicking regular user behavior or using encrypted communication). This can result in false negatives, where genuine threats go undetected.
   Impact: Missed anomalies can lead to security breaches, where attackers exploit vulnerabilities without triggering any alerts.
3. Limited visibility into encrypted traffic
   Explanation: A growing amount of network traffic is encrypted, making it difficult for traditional firewalls and anomaly detection systems to inspect the contents of packets. Without the ability to decrypt traffic, these systems can only rely on metadata such as packet size or traffic patterns, which may not be sufficient to detect anomalies.
   Impact: Anomalies in encrypted traffic may be missed, reducing the effectiveness of intra-firewall detection in identifying modern attacks that use encryption to evade detection.
4. Difficulty in defining "normal" behavior
   Explanation: Establishing a baseline for what is considered "normal" behavior in a network is challenging, especially in dynamic environments where traffic patterns, applications, and services change frequently. Firewalls often lack the context needed to accurately define what is normal, leading to ineffective detection.
   Impact: Systems may either become too sensitive (leading to false positives) or too lenient (leading to false negatives) if the baseline is not well-defined.
5. Lack of contextual awareness
   Explanation: Traditional firewalls focus primarily on packet-level information, such as source and destination IP addresses, ports, and protocols. This limited view often lacks the broader context, such as user behavior, application-level data, or network-wide traffic patterns, which are crucial for accurately detecting anomalies.
   Impact: Firewalls may struggle to detect more sophisticated attacks that blend in with normal traffic patterns, especially those operating at higher levels of the network stack (e.g., application or user level).
6. Difficulty in handling large and complex networks
   Explanation: In large or distributed networks, such as in multi-cloud or hybrid environments, traffic patterns can be highly complex. Firewalls in such environments must process vast amounts of data, which increases the difficulty of detecting subtle anomalies. Network changes, such as scaling of services or introduction of new devices, can further complicate anomaly detection.

Impact: The complexity and volume of data can lead to slower detection or missed anomalies, especially when traffic spans multiple interconnected environments.

7. Limited automation and response capabilities
   Explanation: Many intra-firewall anomaly detection systems generate alerts but require manual intervention to address potential threats. As cyberattacks become more sophisticated and rapid, the need for immediate response becomes more critical.
   Impact: Delays in manual response to detected anomalies can lead to security breaches. Automation capabilities for anomaly detection are often limited, reducing the ability to respond to threats in real-time.

7. RESULTS AND DISCUSSION

The section looks at implementation specifics and the effectiveness of the suggested firewall rule anomaly detection. One host and virtual machines are created and managed using cloudsim, a cloud simulator, and it is developed using the Java platform. The system specs include an Intel Pentium 2.30 GHz CPU, 4 GB of RAM, and Windows 10 64-bit.

This paper suggests an anomaly resolution method based on firewall trees. The following are the crucial steps: When creating a tree, a policy is converted into a corresponding firewall tree by use of a similar firewall tree creation technique, and the rule sections that overlap are noted. A path within a subtree is established using the corresponding tree. The subtree path is used to identify the appropriate anomalous rules [20], [21]. Figure 3 depicts proposed architecture of our tree rule firewall approach. Figure 4 shows running code in netbeans.

When a traffic packet satisfies certain requirements, the administrator can identify it. Which policy decides whether to accept or reject data packets will be made clear in the report. When a new rule is created, the administrator may check to see if it differs from the previously established policy by comparing it to the related tree [22].

We put the suggested methods and strategies into practice in a software tool prototype. This prototype consists of two modules and was created in Java for portability. The configuration of the integrated firewalls and a description of the network topology are read in the first module. Next, using the definitions from the preceding sections, it does a preliminary analysis to gather all the abnormalities among the firewall rules [23].

In order to lower the overall number of anomalies that need to be solved, the second module analyses the entire collection of anomalies and employs the optimal resolution approach. It is expected that the administrator accepts the present state of affairs and abnormalities are removed.

These experiments were conducted to evaluate the methodology's viability in two key areas: identifying abnormalities and eliminating them, even in a tool prototype. The performed experiments yielded early findings that validated the approach's practicality when applied to real-world scenarios. The technique has been beneficial as seen by the reduction in the total anomaly to be evaluated in all trials compared to the initial set [24].

Figure 5 shows the memory utilization after using 100 firewall rules. We have developed upto 1200 rules. Figure 6 shows implemetation of running firewall tree in Cloudsim. Figure 7 shows Firewall execution time in Windows platform and Figure 8 shows when we take 10 rules of firewall as input. Figure 9 shows redundant and shadowing anomailes removal from firewall rules removing inconsistencies in firewall rules. Figure 10 shows time required to remove anomailes in milliseconds.

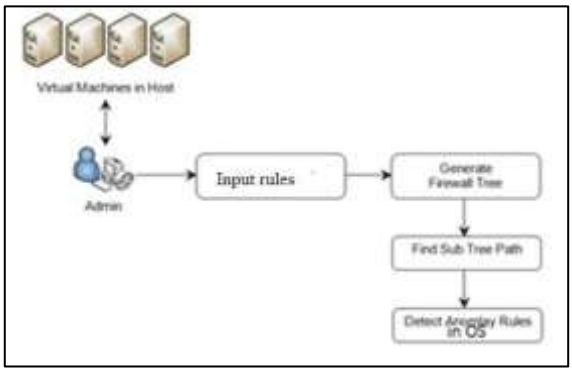

Figure 3. Proposed architecture

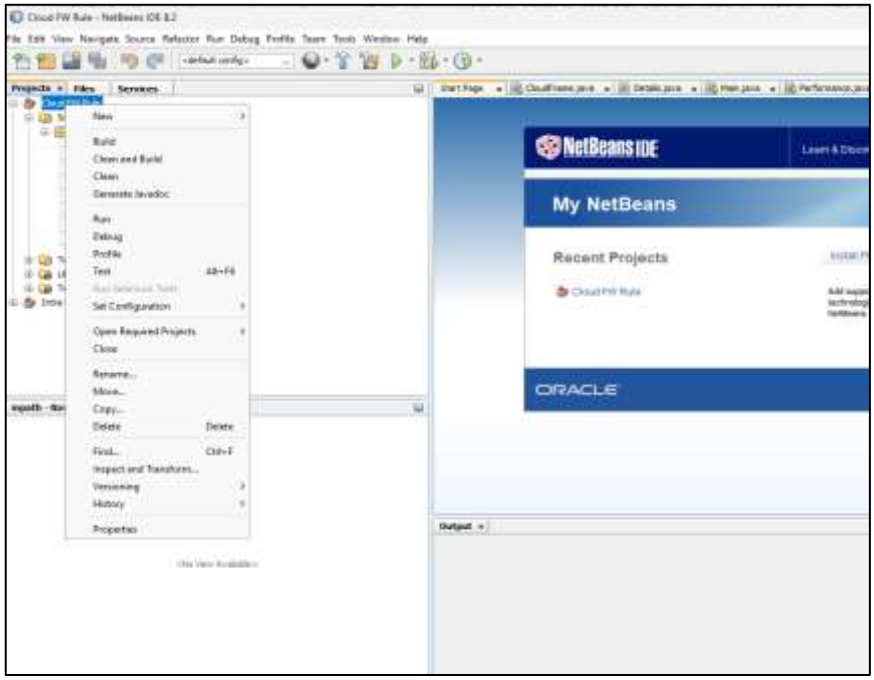

Figure 4. Running cloud sim code in Netbeans

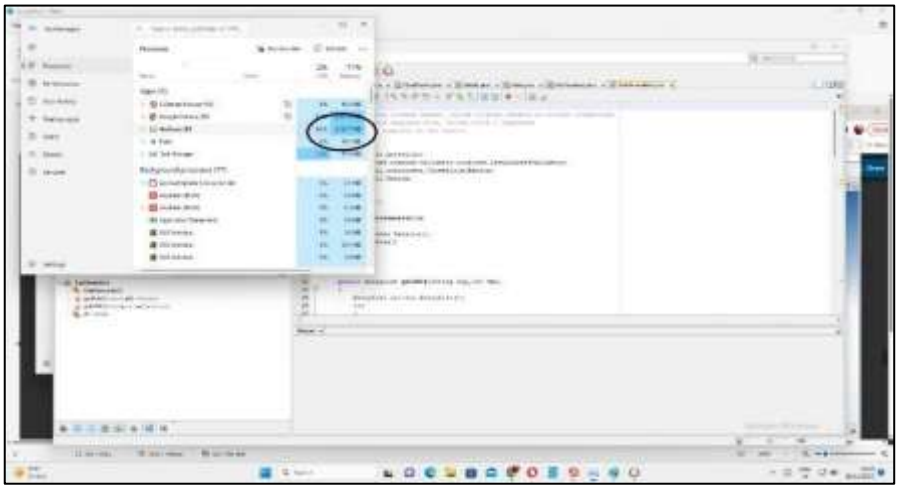

Figure 5. Memory utilization using 100 firewall rules

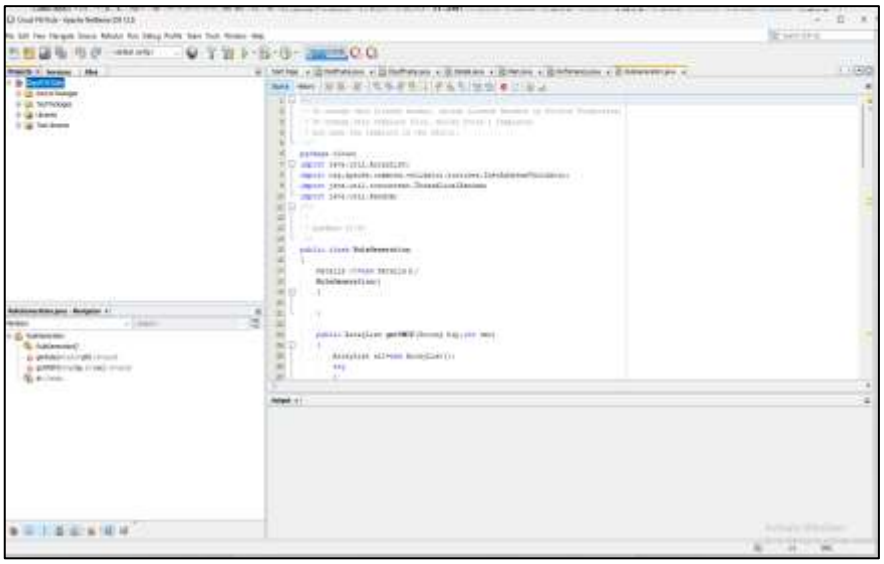

Figure 6. Running tree rule firewall

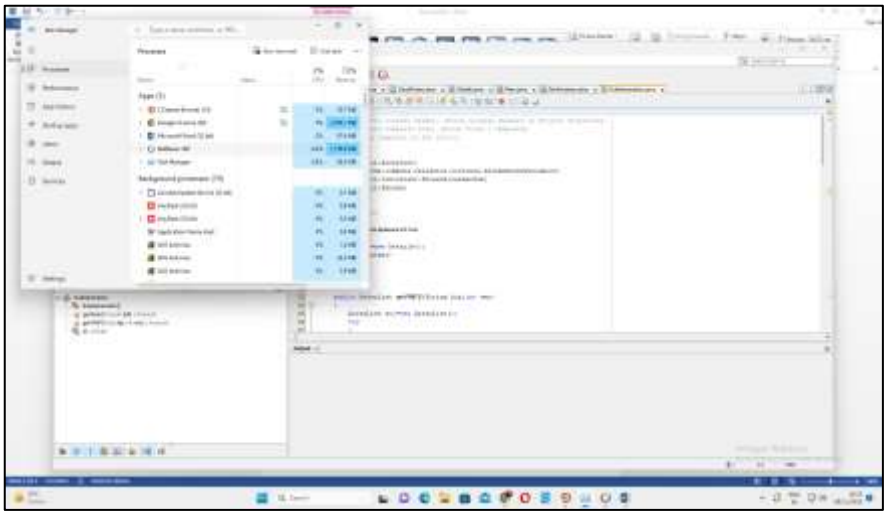

Figure 7. Firewall execution time in Windows platform

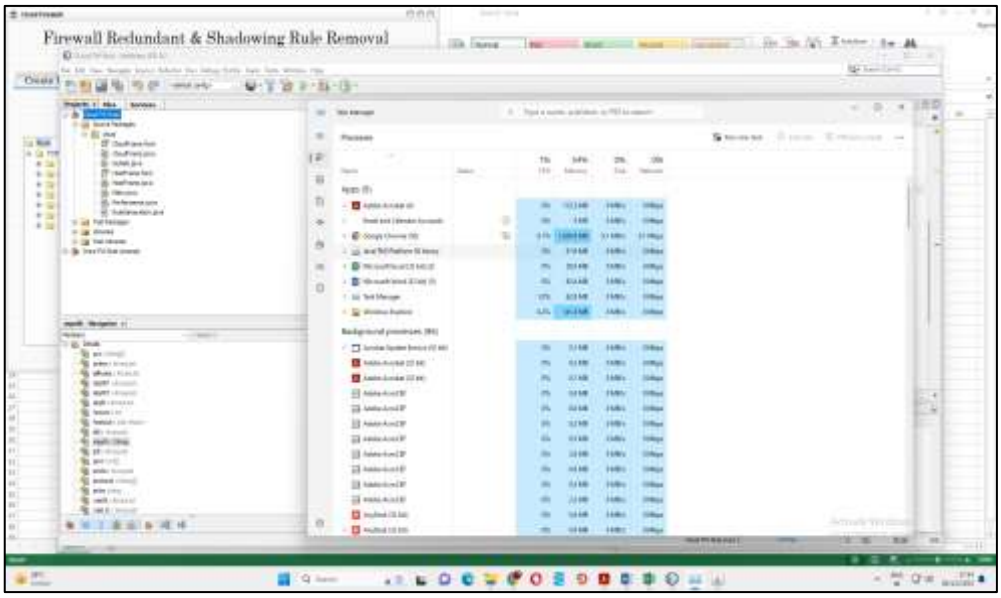

Figure 8. 10 rules as input

Figure 9. Anomalies removal from firewall rules

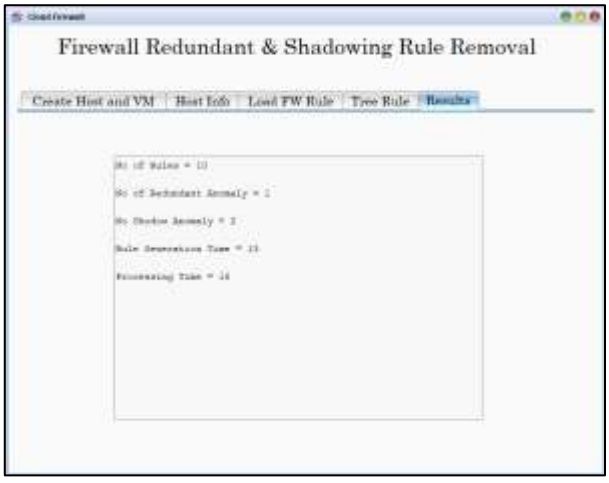

Figure 10. Time required to remove anomalies

8. INDEPTH ARCHITECTURE

This study proposes a firewall tree-based anomaly detection method. The crucial actions are as follows:

The original policy is transformed into an analogous firewall tree utilising a comparable firewall tree generation technique, and the overlapping sections of the rules are saved throughout the tree's creation. The corresponding tree is used to generate a path inside a subtree. The corresponding anomalous rules are found based on this subtree path. Figure 11 describes proposed architecture of our model as tFtree rule firewall.

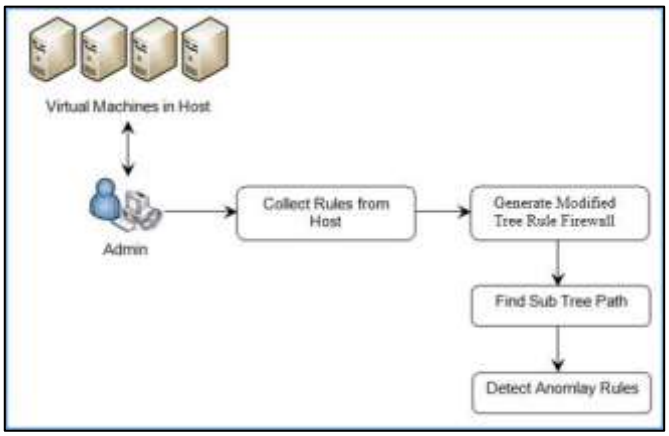

Figure 11. Proposed architecture in detail

Traffic packets that meet several rules can be identified by the administrator. The report will specify which rule will determine whether the data packets are allowed or refused. By comparing a newly added rule with the matching tree, the administrator can ascertain whether the rule deviates from the original policy.

The tree model, which can also be used to rapidly detect relationships and rule irregularities, provides a straightforward representation of the firewall rules. Every level on a node in the structure of the tree represents a possible value for the related field, and each node itself represents an arbitrary conditional filter field. The route of the tree from the root to the focal point at every leaf represents the legal part of each policy regulation.

The firewall tree has the properties shown below.

- The root node is the node that will not have any incoming edges.
- A leaf node is a node without any edges.
- Every node possesses a set of qualities.
- Every edge is a collection on non-empty sets that make up a subset of the area of the parent node.
- The guided path that joins the root and leaf nodes is referred to as a "tree path".

A private cloud is built using open-source software, and firewall rules are managed using the Tree-Rule approach. With this method, packets are tree-likely filtered based on characteristics such protocols and IP addresses. Furthermore, the rate at which packet are screened and processed has been increased in order to keep the virtual firewalls from overloading in this specific scenario.

No redundant rules are added by the modified tree rule firewall (MTRF cloud). It uses a whole network to disperse processing traffic rather than relying just on one cluster or server. In fact, the distributed firewall is designed to defend networks against harmful internal activities, such as those that attempt to compromise VPN or IPsec protocols. For further security and high network traffic throughput, a distributed firewall can be set up in addition to regular firewalls [25], [26].

Organisations are at danger from cyberthreats including ransomware, phishing, hacking, data leaks, and insider threats. This indicates that procedures must be used to safeguard the usefulness and integrity of data.

9. FIREWALL RULES POLICY EDITING

Network administrators frequently write firewall policies, which are then periodically modified (by adding, changing, or deleting rules) to take new security needs and network structure changes into account. It can be significantly more difficult to edit a corporate security policy than to create a new one.

Since a new screening rule might not be applicable to all network sub-domains, it is important to ensure that it is appropriately placed in the appropriate firewalls to prevent allowing or blocking unwanted traffic. Furthermore, because a local firewall policy's rules are arranged in a certain order, adding a new rule requires careful consideration to prevent intra-firewall anomalies.

The same holds true if the regulation is changed or eliminated. This section covers firewall policy editing methods that prevent anomalies from being introduced by policy changes and greatly simplify rule editing work. In order to prevent intra- and inter-firewall anomalies, the policy editor assists the user in identifying the appropriate firewalls to place new rules within and in locating them there. It also offers visual aids to help users monitor and validate policy changes. Administrators may add, change, and delete rules from the firewall policy without any prior expertise or understanding by using the policy editor.

– Rule insertion

There are two phases involved in adding a new regulation to the global security policy. Finding the firewalls where the rule should be installed is the first step. This is necessary to ensure that the filtering rule is only applied to the pertinent sub-domains and that no abnormalities arise between the firewalls. Identifying the correct sequence for the newly implemented rule in each of these firewalls is the second stage in preventing the creation of intra-firewall anomalies.

Generally speaking, any rule that matches a superset should come before this rule, and any rule that matches this rule's subset should come after it. The regional policy tree is employed to identify any possible anomalies and maintain track of the new rule's proper ordering. By comparing each field of the new rule to the relevant tree branch values, we first look for the proper rule location in the policy tree. The sequence of values of the rule that has been added thus far is less than the bare minimum number for all the regulations in that branch if the field result is a portion of the branch.

The order of the most recent rule thus far is higher than the highest order for each of the rules that exist in this branch of rules if the field result is a combination of the branch. Conversely, in the event that a rule is discontinuous, it can grant any order inside the policy. Similar to this, so long as the field's value matches the branch exactly or matches a portion of it, the tree browsing process keeps assessing the subsequent fields in a rule recursively. The rule is entered and given the order decided upon during the browsing phase when it reaches the action field. Every time a disjunct or superset matching is discovered, a new branch is made for the new rule. The policy writer rejects the new rule and informs the individual with a suitable notification if it is duplicate because it is a precise match or a portion of the match and performs the same action as an existing rule.

– Rule removal

An inter-firewall anomaly may be created in distributed firewall setups when a rule is removed from a particular firewall. For instance, removing a "deny" rule within an upstream firewall will cause spurious traffic to flow downstream; however, removing a "accept" rule will block the appropriate traffic and cause all downstream rules that are related (exact, subset, or superset) to be shadowed.

To find the network path between each source-destination domain pair that is important to this rule, we employ the same method that was detailed in the rule insertion procedure. The rule is deleted from the firewall's policy in the following manner in the second phase. We delete the rule from the firewalls in all pathways from source to destination if it is a "accept" rule. If the rule is deleted from the upstream or downstream firewalls, respectively, shadowing or spuriousness anomaly will result. On the other hand, since a "deny" rule has no impact across other firewall in the network, we simply remove the rule from the current firewall in this case.

Another crucial action in the firewall policy is modifying a rule. Nonetheless, using the methods for rule deletion and insertion covered in this section, a changed rule may be simply checked and added.

10. PSEUDOCODE IN DETAIL

This study proposes a firewall tree-based anomaly detection method. The crucial actions are as follows:

Using a matching firewall tree production approach, the policy is transformed into an analogous firewall tree, and the overlapping parts of the regulations are saved during the tree creation process. The corresponding tree is used to generate a path inside a subtree. The corresponding anomalous rules are found based upon the subtree path.

Traffic packets that meet several rules can be identified by the administrator. The report will specify which policy determines whether the data packet are allowed or refused. The administrator has the ability to compare a newly introduced rule to the matching tree and determine whether it deviates from the original policy. The firewall rules are represented by the tree model, which can additionally be used to swiftly spot irregularities in the rules and relationships. The conditional element is shown by the node. Each level of the

node represents a possible value for the corresponding element. The route of the tree from its base through the point where it reaches every leaf represents the legal part of each policy regulation. Algorithm 1 takes firewall rules as input and provides modified tree rule firewall as output.

Algoritma 1. Firewall rules

```
Input: Firewall Rules (FR) {F1,F2,F3,…Fn}
Output: Firewall Tree (FT)
Step1:  If FT is empty, then
Step2:     Add F1 into FT
Step3:  Else
Step4:     For i = 2 to n
Step5:         For j = 1 to m
Step6:            Append (nodej)
Step7:         End For
Step8:     End For
Step9:   EndIf
Step10: Append (node N):
Step11:    e = Edge(N)
Step12:     index=-1
Step13:     while (e ≠ Ø and index<0) do
Step14:            next_E=getNext(e)
Step15:            if next_E ≠ Ø then
Step16:                 index = NodeIndex(next_E)
Step17:            EndIf
Step18:     EndWhile
Step19:     If index < 0
Step20:         insert N into FT
Step21:    EndIf
```

11. CONCLUSION

Firewalls require adequate management in order to provide efficient security services. However, the network could grow dangerous as a result of network vulnerabilities caused by firewall rule intricacy and rule abnormalities. The anomaly finding strategy is used in this research to uncover irregularities in a firewall policy. This study creates a firewall tree model in order to detect firewall anomalies. To locate and simplify the policy, the administrator might utilize the firewall anomaly identification technique. This research will define rule abnormalities in a certain host firewall. We have concluded that as number of firewall rules increases there is increase in memory required of the system. Because firewall policies anomalies can generate better judgements without human involvement and enable effective analysis on a bigger scale, machine learning (ML) and deep learning (DL) approaches have been invaluable tools for years. Optimizing firewall rules within an internal network (intra-firewall optimization) can offer several advantages. By streamlining and fine-tuning firewall rules, organizations can enhance network security, performance, and manageability.

REFERENCES

[1] S. Khummanee, "The semantics loss tracker of firewall rules," in *Advances in Intelligent Systems and Computing*, vol. 769, 2019, pp. 220–231.

[2] A. X. Liu, A. R. Khakpour, J. W. Hulst, Z. Ge, D. Pei, and J. Wang, "Firewall fingerprinting and denial of firewalling attacks," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 7, pp. 1699–1712, Jul. 2017, doi: 10.1109/TIFS.2017.2668602.

[3] M. Cheminod, L. Durante, L. Seno, and A. Valenzano, "An algorithm for security policy migration in multiple firewall networks," *CEUR Workshop Proceedings*, vol. 2940, pp. 344–359, 2021.

[4] A. A. Jabal *et al.*, "Methods and tools for policy analysis," *ACM Computing Surveys*, vol. 51, no. 6, pp. 1–35, Nov. 2019, doi: 10.1145/3295749.

[5] J. Ullrich, J. Cropper, P. Frühwirt, and E. Weippl, "The role and security of firewalls in cyber-physical cloud computing," *EURASIP Journal on Information Security*, vol. 2016, no. 1, p. 18, Dec. 2016, doi: 10.1186/s13635-016-0042-3.

[6] H. Toumi, F. Z. Fagroud, A. Zakouni, and M. Talea, "Implementing Hy-IDS, mobiles agents and virtual firewall to enhance the security in IaaS cloud," *Procedia Computer Science*, vol. 160, pp. 819–824, 2019, doi: 10.1016/j.procs.2019.11.005.

[7] E. Al-Shaer, "Classification and discovery of firewalls policy anomalies," in *Automated Firewall Analytics*, Cham: Springer International Publishing, 2014, pp. 1–24.

[8] M. Abbas, K. Ali, A. Jamali, K. A. Memon, and A. A. Jamali, "Multinomial naive bayes classification model for sentiment analysis," *IJCSNS International Journal of Computer Science and Network Security*, vol. 19, no. 3, pp. 62–67, 2019, [Online]. Available: https://www.researchgate.net/publication/334451164.

[9] S. Khummanee, P. Chomphuwiset, and P. Pruksasri, "DSSF: decision support system to detect and solve firewall rule anomalies based on a probability approach," *ECTI Transactions on Computer and Information Technology (ECTI-CIT)*, vol. 16, no. 1, pp. 56–73, Mar. 2022, doi: 10.37936/ecti-cit.2022161.246996.

[10] H. Hu, G.-J. Ahn, and K. Kulkarni, "Detecting and resolving firewall policy anomalies," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 3, pp. 318–331, May 2012, doi: 10.1109/TDSC.2012.20.

[11] H. Hamed, E. Al-Shaer, and W. Marrero, "Modeling and verification of IPSec and VPN security policies," in *13TH IEEE International Conference on Network Protocols (ICNP'05)*, 2005, vol. 2005, pp. 259–278, doi: 10.1109/ICNP.2005.25.

[12] Y. Yang and C. Science, "Ipsec / Vpn security policy engineering : automatic generation and conflict detection," no. June, 2006.
[13] E. Al-Shaer and S. Al-Haj, "FlowChecker," in *Proceedings of the 3rd ACM workshop on Assurable and usable security configuration*, Oct. 2010, pp. 37–44, doi: 10.1145/1866898.1866905.
[14] A. D. Ferguson, A. Guha, C. Liang, R. Fonseca, and S. Krishnamurthi, "Hierarchical policies for software defined networks," in *Proceedings of the first workshop on Hot topics in software defined networks*, Aug. 2012, pp. 37–42, doi: 10.1145/2342441.2342450.
[15] C. Togay, A. Kasif, C. Catal, and B. Tekinerdogan, "A firewall policy anomaly detection framework for reliable network security," *IEEE Transactions on Reliability*, vol. 71, no. 1, pp. 339–347, Mar. 2022, doi: 10.1109/TR.2021.3089511.
[16] P. Wang, L. Huang, H. Xu, B. Leng, and H. Guo, "Rule anomalies detecting and resolving for software defined networks," in *2015 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2014, pp. 1–6, doi: 10.1109/GLOCOM.2014.7417386.
[17] F. A. Maldonado-Lopez, E. Calle, and Y. Donoso, "Detection and prevention of firewall-rule conflicts on software-defined networking," in *2015 7th International Workshop on Reliable Networks Design and Modeling (RNDM)*, Oct. 2015, pp. 259–265, doi: 10.1109/RNDM.2015.7325238.
[18] S. Pisharody, A. Chowdhary, and Dijiang Huang, "Security policy checking in distributed SDN based clouds," in *2016 IEEE Conference on Communications and Network Security (CNS)*, Oct. 2016, pp. 19–27, doi: 10.1109/CNS.2016.7860466.
[19] E. S. Al-Shaer and H. H. Hamed, "Discovery of policy anomalies in distributed firewalls," in *IEEE INFOCOM 2004*, 2004, vol. 4, pp. 2605–2616, doi: 10.1109/INFCOM.2004.1354680.
[20] Y.-Z. Cheng and Q. Shi, "Analysis of policy anomalies in distributed firewalls," *International Journal of Network Security*, vol. 24, no. 4, pp. 617–627, 2022.
[21] G. Jekese and R. Subburaj, "Virtual firewall security on virtual machines in cloud environment," *International Journal of Scientific & Engineering Research*, vol. 6, no. 2, 2015, [Online]. Available: http://www.ijser.org.
[22] N. Dezhabad and S. Sharifian, "Learning-based dynamic scalable load-balanced firewall as a service in network function-virtualized cloud computing environments," *The Journal of Supercomputing*, vol. 74, no. 7, pp. 3329–3358, Jul. 2018, doi: 10.1007/s11227-018-2387-5.
[23] S. Bagheri and A. Shameli-Sendi, "Dynamic firewall decomposition and composition in the cloud," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3526–3539, 2020, doi: 10.1109/TIFS.2020.2990786.
[24] J. J. Praise, R. J. S. Raj, and J. V. B. Benifa, "Development of reinforcement learning and pattern matching (RLPM) based firewall for secured cloud infrastructure," *Wireless Personal Communications*, vol. 115, no. 2, pp. 993–1018, Nov. 2020, doi: 10.1007/s11277-020-07608-4.
[25] P. R. Kadam and Bhusari, "Redundancy removal of rules with reordering them to increase the firewall optimization," *International Journal of Research in Engineering and Technology*, vol. 03, no. 10, pp. 317–321, Oct. 2014, doi: 10.15623/ijret.2014.0310051.
[26] Z. Lin and Z. Yao, "Firewall anomaly detection based on double decision tree," *Symmetry*, vol. 14, no. 12, p. 2668, Dec. 2022, doi: 10.3390/sym14122668.

BIOGRAPHIES OF AUTHORS

Mrs. Dhwani Rajkumar Hakani is a full time Ph.D. Scholar at Gujarat Technological University School of Engineering and Technology. She has a total 7 years and 6 months of experience. She is associated with GTU since 20th April, 2015. Prior to join GTU-GSET, she had worked at Gujarat Technological University as a Research Assistant in IT department from April 2015 to April 2022. She had been handling various GTU portals and Cloud management in IT section of GTU.She had completed her Graduation and Post-Graduation both from L.J Institute of Engineering and Technology College. She has done her dissertation research work in the area of mobile cloud computing. Her area of interest is cloud computing and security. She has published 1 survey paper and 1 research paper in various international journals in the area of mobile cloud computing. She can be contacted at email: adf_dhwani@gtu.edu.in.

Dr Palvinder Singh Mann received his Bachelor's Degree (B.Tech) with Honours (Institute Gold Medal) in Information Technology from Kurukshetra University, Haryana, India, Master's Degree (M.Tech) in Computer Science and Engineering and Ph.D. in Computer Science and Engineering from IKG Punjab Technical University, Punjab, India. He has more than 19 years' experience in academics and research and presently, working as Associate Professor at Gujarat Technological University, Ahmedabad, Gujarat, INDIA. He has published more than 80 research papers in various International Journals, and Conferences. He is reviewer of many distinguished journals published by IEEE, Elsevier and Springer. He has delivered experts talks at various organizations and chaired many conferences as well. His research interest includes artificial intelligence, soft computing and cloud security. He can be contacted at email: asso_psmaan@gtu.edu.in.